

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

1c675 U.S. PTO
09/370152
06/09/99

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1 9 9 8 年 9 月 2 1 日

出 願 番 号

Application Number:

平成 1 0 年 特 許 願 第 2 6 5 9 5 0 号

出 願 人

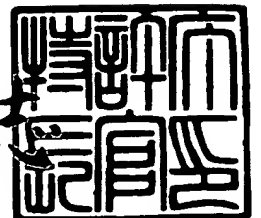
Applicant (s):

株式会社日立製作所

1 9 9 9 年 5 月 1 4 日

特 許 庁 長 官
Commissioner,
Patent Office

伴 佐 山 建 志



出 証 番 号 出 証 特 平 1 1 - 3 0 2 9 1 7 0

【書類名】 特許願

【整理番号】 1198025931

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/16

【発明の名称】 リアルタイム分散システム

【請求項の数】 14

【発明者】

 【住所又は居所】 東京都小平市上水本町五丁目20番1号
株式会社 日立製作所 半導体事業部内

 【氏名】 今井 崇明

【発明者】

 【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社 日立製作所 日立研究所内

 【氏名】 横山 孝典

【発明者】

 【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社 日立製作所 日立研究所内

 【氏名】 永浦 渉

【発明者】

 【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社 日立製作所 日立研究所内

 【氏名】 鈴木 昭二

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社 日立製作所

【代理人】

 【識別番号】 100068504

 【弁理士】

 【氏名又は名称】 小川 勝男

【電話番号】 03-3212-1111
【手数料の表示】
【予納台帳番号】 013088
【納付金額】 21,000円
【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 リアルタイム分散システム

【特許請求の範囲】

【請求項 1】

ネットワーク通信を行うネットワークコントローラと、プログラムを記憶するメモリと、上記メモリに記憶されたプログラムを実行するCPUとを有し、上記メモリに記憶されたプログラムを上記CPUで実行しながら、上記ネットワークコントローラを用いて、通信の優先順位を持ったメッセージの送受信処理を実行するリアルタイム分散システムにおいて、複数の送受信メッセージを格納した前記ネットワークコントローラによって複数のメッセージ送受信処理を実行してネットワーク通信を行うネットワークドライバと、扱う送受信メッセージの通信の優先順位に応じて上記ネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、上記ネットワークドライバの処理の優先順位にしたがって上記ネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システム。

【請求項 2】

ネットワーク通信を行うネットワークコントローラと、プログラムを記憶するメモリと、メモリに記憶されたプログラムを実行するCPUとを有し、上記メモリに記憶されたプログラムを上記CPUで実行しながら、上記ネットワークコントローラを用いて、通信の優先順位を持ったメッセージの送受信処理を実行するリアルタイム分散システムにおいて、複数の送受信メッセージを記憶するメッセージ記憶部と、上記メッセージ記憶部に記憶された複数のメッセージの送受信処理を、上記ネットワークコントローラを用いて、実行してネットワーク通信を行うネットワークドライバと、扱う送受信メッセージの通信の優先順位に応じて上記ネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、上記ネットワークドライバの処理の優先順位にしたがってネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システム。

【請求項 3】

ネットワーク通信を行うネットワークコントローラと、プログラムを記憶するメモリと、メモリに記憶されたプログラムを実行するCPUとを有し、上記メモリに記憶されたプログラムを上記CPUで実行しながら、上記ネットワークコントローラを用いて、通信の優先順位を持ったメッセージの送受信処理を実行するリアルタイム分散システムにおいて、複数の送受信メッセージを記憶するメッセージ記憶部と、前記ネットワークコントローラを用いて、ネットワーク通信を行うネットワークドライバと、上記メッセージ記憶部に記憶された複数のメッセージの送受信処理を、上記ネットワークドライバ手段を用いて実行する通信処理ライブラリと、扱う送受信メッセージの通信の優先順位に応じて上記ネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、上記ネットワークドライバの処理の優先順位にしたがってネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システム。

【請求項 4】

上記ネットワークコントローラは複数のメッセージを記憶するための複数のメールボックスを有し、該複数のメールボックスに記憶されたメッセージを、上記メッセージの優先順位に応じてネットワーク通信を実行できる請求項 1、請求項 2 または請求項 5 記載のリアルタイム分散システム。

【請求項 5】

上記複数のメールボックスに対して、それぞれひとつの優先順位のメッセージを記憶する請求項 4 記載のリアルタイム分散システム。

【請求項 6】

上記複数のメールボックスのひとつに対して、複数の優先順位のメッセージを記憶する請求項 4 記載のリアルタイム分散システム。

【請求項 7】

上記ネットワークコントローラは、CAN プロトコルを処理するCAN コントローラであり、上記通信の優先順位は上記メッセージのメッセージIDで指定される請求項 1、請求項 2 または請求項 3 記載のリアルタイム分散システム。

【請求項 8】

上記ネットワークドライバ手段は、メッセージ送信処理を行うメッセージ送信タスクと、メッセージ受信処理を行うメッセージ受信タスクとを有し、上記スケジューリング部は、上記ネットワークドライバの処理の優先順位にしたがって、上記メッセージ送信タスクとメッセージ受信タスクを実行する請求項 1、請求項 2 または請求項 3 記載のリアルタイム分散システム。

【請求項 9】

上記メッセージ記憶部は、OSEK-COM プロトコルのメッセージを記憶する、メッセージオブジェクトである請求項 2 または請求項 3 記載のリアルタイム分散システム。

【請求項 10】

前記ネットワークドライバは、送信メッセージまたは受信メッセージを一時記憶するメッセージバッファを有し、送信したメッセージまたは受信したメッセージは一旦該メッセージバッファに記憶した後、上記メッセージ記憶手段に記憶する請求項 3 乃至請求項 5 記載のリアルタイム分散システム。

【請求項 11】

上記ネットワークドライバ手段は、送信メッセージまたは受信メッセージを一時記憶するメッセージバッファを有し、上記メールボックスのひとつに記憶されるメッセージは、上記メッセージバッファのひとつに記憶する請求項 4 乃至請求項 6 記載のリアルタイム分散システム。

【請求項 12】

上記ネットワークドライバ手段は、送信メッセージまたは受信メッセージを一時記憶するメッセージバッファを有し、上記メールボックスの複数に記憶されるメッセージを、上記メッセージバッファのひとつに記憶する請求項 4 乃至請求項 6 記載のリアルタイム分散システム。

【請求項 13】

前記通信処理ライブラリ手段は、OSEK-COM プロトコルにしたがい、メッセージ送信処理と、メッセージ受信処理を行うことを特徴とする請求項 3 記載のリアルタイム分散システム。

【請求項 14】

前記ネットワークドライバ手段は、メッセージ送信処理を行うメッセージ送信タスクと、メッセージ受信処理を行うメッセージ受信タスクとを含み、前記スケジューリング手段は、前記ネットワークドライバの処理の優先順位にしたがって、前記メッセージ送信タスクとメッセージ受信タスクを、前記通信処理ライブラリ手段のメッセージ送信処理やメッセージ受信処理を実行するタスクとは別のタスクとして実行することを特徴とする請求項 13 に記載のリアルタイム分散システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はネットワーク接続されたリアルタイム分散システムに係り、特に、メッセージの送受信処理を、メッセージが与えられた優先度に従って処理を行うのに好適なリアルタイム分散システムに関する。

【0002】

【従来の技術】

ネットワークによりメッセージを送受信する場合、ネットワーク通信をハードウェアで実現するネットワークコントローラと、ネットワークコントローラを用いてネットワーク通信を行うネットワークドライバを用いる。いま、ネットワークコントローラに設けられた送受信メッセージを格納するバッファをメールボックスと呼ぶことにする。例えば送信の場合、アプリケーションからネットワークドライバへの送信要求は送信要求キューに並べられる。ネットワークドライバは送信要求キューから送信要求を 1 つずつ取り出して、送信メッセージをメールボックスへ格納してネットワーク送信処理を行う。通常、送信要求キューは FIFO でキューイングされるため、送信要求時間順にネットワーク送信処理が行われる。しかし、リアルタイム分散システムでは、メッセージに優先順位を付加し、優先順位に応じたネットワーク通信処理を行うことが望まれる。そこで、例えば、CQ 出版社刊、インターフェース（1994 年、第 12 号）、72 頁～148 頁に記載されているように、キューイングされている送信要求をアプリケーション

が送信するメッセージの優先順位にしたがって入れ替えることにより、ネットワーク通信の優先順位制御を行う方式がある。これにより、ネットワークドライバは送信要求時間順ではなく、ネットワークドライバが送信要求キューから送信要求を取り出す時点での送信データの優先順位にしたがってネットワーク送信処理が行える。

【0003】

【発明が解決しようとする課題】

しかしながら上記従来技術では、一旦ネットワークドライバが起動されて、ネットワーク送信処理が開始されると、現在ネットワーク送信処理中のデータよりも高い優先順位のメッセージに対して送信要求がなされても、現在のネットワーク送信処理が完了するまで、高い優先順位のメッセージはネットワーク送信処理が行えないという問題がある。

【0004】

例えば、ネットワークコントローラが複数接続されたリアルタイム分散システムを考える。このリアルタイム分散システムは、メッセージに付加された優先順位によりネットワークの優先順位制御を行うネットワークを用いるとする。複数のネットワークコントローラがネットワークへメッセージを送信する場合、メッセージに付加された優先順位によりネットワークの優先順位制御が行われ、送信要求のなされたメッセージで最も高い優先順位のメッセージの送信要求を行ったネットワークコントローラがネットワークへメッセージを送信することができる。このとき、低い優先順位のメッセージの送信要求を行ったネットワークコントローラはネットワークへのメッセージ送信が遅れてしまう。このため、低い優先順位のメッセージのネットワーク送信処理が完了するまで、このネットワークコントローラが高い優先順位のメッセージのネットワーク送信処理を行えない場合、高い優先順位のメッセージのネットワーク送信処理はさらに遅れ、システムのリアルタイム性の保証に問題を生じる。

【0005】

そこで本発明は、ネットワークドライバ処理の開始後においても送信要求のなされたメッセージで最も優先順位の高いメッセージを送信できる分散処理システム

ムを提供することを目的とする。

【0006】

【課題を解決するための手段】

上記目的は、複数の送受信メッセージを格納したネットワークコントローラによって複数のメッセージ送受信処理を実行してネットワーク通信を行うネットワークドライバと、扱う送受信メッセージの通信の優先順位に応じてネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、ネットワークドライバの処理の優先順位にしたがって上記ネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システムによって達成することができる。

【0007】

また、上記目的は複数の送受信メッセージを記憶するメッセージ記憶部と、メッセージ記憶部に記憶された複数のメッセージの送受信処理を、ネットワークコントローラを用いて、実行してネットワーク通信を行うネットワークドライバと、扱う送受信メッセージの通信の優先順位に応じてネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、ネットワークドライバの処理の優先順位にしたがってネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システムによって達成することができる。

【0008】

また、複数の送受信メッセージを記憶する記憶部と、ネットワークコントローラを用いて、ネットワーク通信を行うネットワークドライバと、メッセージ記憶部に記憶された複数のメッセージの送受信処理を、ネットワークドライバ手段を用いて実行する通信処理ライブラリと、扱う送受信メッセージの通信の優先順位に応じてネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理部と、ネットワークドライバの処理の優先順位にしたがってネットワークドライバの処理を実行するスケジューリング部とを有するリアルタイム分散システムによって達成することができる。

【0009】

このような構成により、一旦ネットワークドライバが起動されてメッセージ送信処理のタスクが開始されても、現在送信処理中のメッセージよりも高い優先順位のメッセージに対して送信要求がなされた場合は、現在のネットワーク送信処理完了を待たずに、並列により優先順位の高いネットワークドライバの送信処理タスクを起動して、優先順位の高いメッセージの送信処理を先に行う。これにより、送信要求のなされたメッセージで最も優先順位の高いメッセージを送信できる。受信についても同様に、あるメッセージ受信タスクの処理中に、より優先順位の高いメッセージを受信したときは、より優先順位の高いネットワークドライバの受信処理タスクを起動して、先に受信処理を行う。これにより最も優先順位の高いメッセージから順に受信できる。

【0010】

【発明の実施の形態】

(1) 本発明の一実施例

(a) 本発明の一実施例の全体構成

図1は本発明の一実施例であるリアルタイム分散システムの構成を示したものである。リアルタイム分散システムは制御ユニット2000および2060がネットワーク2005で接続されている。本発明のネットワークは、例えば、オープン・デバイスネット・ベンダー・アソシエーション (Open DeviceNet Vendor Association) 社発行、デバイスネット (DeviceNet) 仕様書 (ボリューム1 (Volume 1), リリース2.0 (Release2.0), 1998年)、第2章2.1～2.8に記載されているCAN (Controller Area Network) を用いる。制御ユニット2000は、CPU2001, メモリ2002, CANコントローラ2003で構成され、それらはバス2004で接続されている。CANコントローラ2003は、ネットワーク2005を介して、他の制御ユニット2060に内蔵されるCANコントローラと接続されている。メモリ2002には、アプリケーションプログラム2006, OSEK-COM2008, CANDライバ2009, OS2050が記憶されている。ここで、OSEK-COMは、オー・エス・イー・ケイ (OSEK) 発行、オー・エス・イー・ケイ/ブイ・ディー・エックス

コミュニケーション (OSEK/VDX Communication) 仕様書 (バージョン 2.1 (Version2.1), リビジョン 1 (revision1)、1998 年) に記載されている OSEK-COM プロトコルを処理するプログラムである。CPU2001 は、メモリ 2002 に記憶されているプログラムを読み出して、プログラム処理を実行する。OS2050 は複数のタスクを並列して実行できるマルチタスク機能を持つ。CAN コントローラ 2003 は、ネットワーク 2005 を介して、他の制御ユニット 2060 とのメッセージ送信、メッセージ受信を行うハードウェアである。

以下、CAN コントローラ 2003, アプリケーションプログラム 2006, OSEK-COM2008, CAN ドライバ 2009, OS2050 の順に詳細に説明する。

【0011】

(b) CAN コントローラ

まず、CAN コントローラについて詳細に説明する。CAN コントローラ 2003 が使用する通信プロトコルは CAN プロトコルである。CAN プロトコルは、各メッセージに固有なメッセージ ID を持たせる。メッセージ ID には優先順位が割り当てられ、ネットワーク 2005 の優先順位制御 (バスアービトレーション) はメッセージ ID の優先順位に基づいて行われる。メッセージ ID によるバスアービトレーションについて説明する。例えば、制御ユニット 2000 が、ネットワーク 2005 を介して、制御ユニット 2000 と同様な構成を持つ、複数の制御ユニットに接続された場合を考える。各制御ユニットが同時に、ネットワーク 2005 へメッセージを送信する場合、各メッセージの中で、最も優先順位の高いメッセージ ID を持つメッセージを送信する制御ユニットだけが、ネットワーク 2005 の使用权を獲得して (バスアービトレーションに勝つ)、優先的にネットワーク 2005 へメッセージを送信することができる。

【0012】

次に CAN コントローラの構成について詳細に説明する。CAN コントローラ 2003 は、コントロールレジスタ 2022, 送信・受信メールボックス 2023, 優先順位制御部 2024, 送信バッファ 2025, 受信フィルタ 2027, 受信バッファ 2026 で構成される。以下、これらについて詳細に説明する。

【0013】

送信・受信メールボックス2023について説明する。送信・受信メールボックス2023は、メールボックスa2028、メールボックスb2029、メールボックスc2030等の複数のメールボックスで構成され、送信メッセージ、受信メッセージを格納する。図2は、メールボックスa2028の詳細な構成図である。メールボックスa2028は、メールボックス番号6004、MessageID6001、DataSize6002、Data6003で構成される。メールボックス番号6004は、送信・受信メールボックス2023の各メールボックスごとに固有に与えられるもので、メールボックス番号を記憶する領域である。

【0014】

MessageID6001は、送信メッセージ、受信メッセージのメッセージIDを記憶する領域である。メールボックスa2028は、MessageID6001を持つ送信メッセージ、受信メッセージのみを格納する。DataSize6002は、格納する送信メッセージ、受信メッセージのデータサイズを記憶する領域である。Data6003は、送信メッセージ、受信メッセージのデータ格納領域である。メールボックスb2029、メールボックスc2030の詳細な構成は、メールボックスa2028と同様である。

【0015】

図3は、コントロールレジスタ2022の詳細な構成図である。コントロールレジスタ2022は、メールボックス番号5001、TXPR5002、TXACK5003、RXPR5004、MBIMR5005で構成される。メールボックス番号5001は、各メールボックスごとに、メールボックス番号を記憶する領域である。TXPR5002は、各メールボックス番号ごとに、メッセージ送信要求を記憶する領域であり、CANコントローラ2003にメッセージ送信を要求するときに、ビットセットする。TXACK5003は、各メールボックス番号ごとに、メッセージ送信完了を記憶する領域であり、CANコントローラ2003がメッセージ送信を完了したときに、ビットセットされる。RXPR5004は、各メールボックス番号ごとに、メッセージ受信完了を記憶する領域であり、CANコントローラ2003がメッセージ受信を完了したときに、ビットセットされる。MBIMR5005は、各メールボックス番号ごと

に、メッセージ送信完了時、または、メッセージ受信完了時の割り込み禁止を記憶する領域であり、メッセージ送信完了時、または、メッセージ受信完了時の、CANコントローラ2003からCPU2001への割り込みを禁止するときに、ビットセットする。

【0016】

優先順位制御部2024は、メッセージ送信要求が発生したときに、対応するメールボックスのメッセージを、送信バッファ2025へ転送する。また、同時に複数のメッセージ送信要求が発生したとき、対応する各メールボックスに格納されているメッセージIDを比較して、最も優先順位の高いメッセージIDを格納しているメールボックスのメッセージを、送信バッファ2025へ転送する。

送信バッファ2025は、優先順位制御部2024から転送されたメッセージを格納する。格納されたメッセージは、メッセージIDによるバスアービトレーションに参加する。メッセージIDによるバスアービトレーションに勝った場合、メッセージはネットワーク2005へ送信される。メッセージIDによるバスアービトレーションに負けた場合、メッセージはネットワーク2005へ送信されず、次のバスアービトレーションの機会を待つ。メッセージIDによるバスアービトレーションに負けた場合、かつ、より優先順位の高いメッセージIDを持つメッセージに対してメッセージ送信要求が発生している場合、優先順位制御部2024は、より優先順位の高いメッセージIDを格納しているメールボックスのメッセージを、送信バッファ2025へ転送する。

【0017】

受信バッファ2026は、ネットワーク2005からの受信メッセージを格納する。

【0018】

受信フィルタ2027は、受信バッファ2026に格納されたメッセージのメッセージIDと、各メールボックスに登録されたメッセージIDを比較して、同一のメッセージIDがあれば対応するメールボックスへメッセージを格納する。

次に、メモリ2002に記憶されている、アプリケーションプログラム2006、OSEK-COM2008、CANドライバ2009、OS2050について詳細に説明する。

【0019】

(c) アプリケーションプログラム

アプリケーションプログラムについて説明する。アプリケーションプログラム2006は、図1に示すように、APa2012, APb2013, APc2014等の複数のアプリケーションプログラムで構成される。本実施例ではこれらのアプリケーションプログラムはそれぞれ別のタスクとして実行される。

【0020】

(d) OSEK-COM

次にOSEK-COMの構成について詳細に説明する。OSEK-COM2008は、メッセージオブジェクトa2015, メッセージオブジェクトb2016, メッセージオブジェクトc2017等の複数のメッセージオブジェクト, メッセージ送信処理2010, メッセージ受信処理2011で構成される。なお、本実施例で用いられるメッセージには、Queuedメッセージ, Unqueuedメッセージの2種類がある。Queuedメッセージは、キューイングが必要で上書き不可能なメッセージである。Unqueuedメッセージは、キューイングが不要で、上書き可能なメッセージである。

【0021】

図4は、メッセージオブジェクトa2015の詳細な構成図を示したものである。メッセージオブジェクトa2015は、sMsgObj3001, Status3007, Data3008, Buffer3009, sMsgNetParams3013, sQueuedMsgInfo3018, sQueuedMsgObjA3022, sQueuedMsgObjB3030等の複数のsQueuedMsgObj, Status3024, Status3032等の複数のStatus, FIFOBufferA3029, FIFOBufferB3037等の複数のFIFOBuffer, NULL3011, NULL3038等の複数のNULL、で構成される。

【0022】

Status3007は、sMsgObjの状態（送信状態、バッファの空き状態）を記憶する領域である。Status3024は、sQueuedMsgObjA3022の状態（バッファの空き状態）を記憶する領域である。Status3032は、sQueuedMsgObjB3030の状態（バッファの空き状態）を記憶する領域である。Data3008は、メッセージのデータを記憶する領域である。Buffer3009は、メッセージがUnqueuedメッセージの場合(3012)

、Data3008のバッファである。FIFOBufferA3029は、メッセージがQueued メッセージの場合（3012）、Data3008のバッファである。FIFOBufferB3037 は、メッセージがQueuedメッセージの場合（3012）、Data3008のバッファである。

sMsgObj3001は、Size3002, StatusRef3003, DataRef3004, sQueuedMsgInfoRef3005, sMsgNetParamsRef3006で構成される。Size3002 は、メッセージのデータサイズを記憶する領域である。StatusRef3003は、Status3007へのポインタである。DataRef3004は、Data3008へのポインタである。

【0023】

sQueuedMsgInfoRef3005は、メッセージがQueuedメッセージの場合、sQueuedMsgInfo3018へのポインタである。sQueuedMsgInfoRef3005 は、メッセージがUnqueuedメッセージの場合、NULL3011へのポインタである。

【0024】

sMsgNetParamsRef3006は、メッセージ送信、または、メッセージ受信が、ネットワーク2005経由の場合、sMsgNetParams3013 へのポインタである。

【0025】

sMsgNetParamsRef3006は、メッセージ送信、または、メッセージ受信が、ネットワーク2005経由でない場合、NULL3011へのポインタである。

【0026】

sMsgNetParams3013は、TransferDir3014, TransferMode3015, TimePeriod3016, Handle3017で構成される。TransferDir3014 は、メッセージが送信メッセージであるか、メッセージが受信メッセージであるかを記憶する領域である。

【0027】

TransferMode3015は、Periodicalメッセージ送信を利用するか、Periodicalメッセージ送信を利用しないかを記憶する領域である。Periodicalメッセージ送信を利用すると、アプリケーションプログラムは周期時間ごとにメッセージ送信が行える。TimePeriod3016は、Periodicalメッセージ送信を利用する場合、その周期時間を記憶する領域である。Handle3017は、メッセージのメッセージIDを記憶する領域である。

【0028】

sQueuedMsgInfo3018は、QueueDepth3019、NRec3020、sQueuedMsgObjRef3021で構成される。QueueDepth3019は、FIFOBufferのキューイング深さつまり、メモリの使用領域を記憶する領域である。NRec3020は、メッセージオブジェクト利用者を記憶する領域である。例えば、APa2012、APb2013の2つのアプリケーションプログラムが、1つのメッセージオブジェクトa2015を利用する場合、NRec3020は2となる。図4は、NRec3020が2の場合を示しており、メッセージオブジェクトa2015は、2つのsQueuedMsgObj(3022, 3030)、2つのStatus(3024, 3032)、2つのFIFOBuffer(3029, 3037)を持つ構成となる。sQueuedMsgObjRef3021は、sQueuedMsgObjへのポインタである。

sQueuedMsgObjの構成について、sQueuedMsgObjA3022を例として説明する。sQueuedMsgObjA3022は、StatusRef3023、BoundaryDataRef3025、WriteDataRef3026、ReadDataRef3027、sQueuedMsgObjRef3028で構成される。

【0029】

StatusRef3023は、Status3024へのポインタである。BoundaryDataRef3025は、FIFOBufferA3029の先頭アドレスである。WriteDataRef3026は、FIFOBufferA3029の書き込みアドレスである。ReadDataRef3027は、FIFOBufferA3029の読み出しアドレスである。sQueuedMsgObjRef3028は、sQueuedMsgObjB3030へのポインタである。sQueuedMsgObjB3030の構成は、sQueuedMsgObjA3022の構成と同様である。ただし、sQueuedMsgObjRef3036はNULL3038へのポインタである。

【0030】

次にOSEK-COMの動作について説明する。まずOSEK-COMのメッセージ送信処理2010について詳細に説明する。メッセージ送信処理2010は、各アプリケーションプログラムが、OSEK-COM2008を用いて実際のメッセージ送信を行うためのプログラムである。本実施例では、OSEK-COMのメッセージ送信処理は、それを呼び出したアプリケーションプログラムのタスクで実行される。図5は、メッセージ送信処理2010の流れを示すフローチャートである。図5を用いて、メッセージ送信処理2010について説明する。例えば、APa2012が、メッセージオブジェクトa2015、メールボックスa2028を

利用して、メッセージ送信を行うとする。

【0031】

まずステップ100で、APa2012は、メッセージ送信で利用するメッセージオブジェクトa2015を決定する。

【0032】

ステップ101では、sMsgNetParamsRef3006を参照して、メッセージ送信がユニット内の通信か、ネットワーク2005経由であるかを判定する。

【0033】

ユニット内の通信の場合、ステップ102で、APa2012が送信するメッセージのデータを、Data3008に書き込む。

【0034】

送信するメッセージがUnqueuedメッセージの場合、Data3008をBuffer3009に書き込む。

【0035】

送信するメッセージがQueuedメッセージの場合、Data3008をFIFOBuffer (3029, 3037) に書き込む。

【0036】

メッセージ送信がユニット内の通信でない場合、すなわち、ネットワーク経由の通信の場合、ステップ103で、TransferDir3014を参照して、送信するメッセージが送信メッセージであるか判定する。

【0037】

次にステップ104で、送信するメッセージがQueuedメッセージであるか判定する。

【0038】

送信するメッセージがQueuedメッセージの場合、ステップ105で、Queuedメッセージ送信処理を行う。

【0039】

送信するメッセージがUnqueuedメッセージの場合、ステップ106で、Unqueuedメッセージ送信処理を行う。

【0040】

図6は、図5のステップ102の処理詳細を示すフローチャートである。図6を用いて、ステップ102を説明する。

【0041】

まずステップ200で、APa2012 が送信するメッセージのデータを、Data3008に書き込む。

【0042】

ステップ201では、送信するメッセージがUnqueuedメッセージの場合、Data3008をBuffer3009に書き込む。

【0043】

送信するメッセージがQueuedメッセージの場合、Data3008をFIFOBuffer (3029, 3037) に書き込む。

【0044】

図7は、図6のステップ201の処理詳細を示すフローチャートである。図7を用いて、ステップ201を説明する。

【0045】

まずステップ300で、送信するメッセージがQueuedメッセージであるか判定する。

【0046】

送信するメッセージがQueuedメッセージである場合、ステップ301で、Data3008をFIFOBuffer (3029, 3037) へ書き込む。

【0047】

送信するメッセージがQueuedメッセージでない場合、まずステップ302でDataRef3004, Size3002 を参照して、Buffer3009の書き込みアドレスを決定する。次にステップ303で、Data3008をBuffer3009へ書き込む。

【0048】

図8は、図7のステップ301の処理詳細を示すフローチャートである。図8を用いて、ステップ301を説明する。

【0049】

まずステップ400で、Size3002を参照して、送信するメッセージのデータサイズを決定する。

【0050】

ステップ401では、DataRef3004を参照して、Data3008 のアドレスを決定する。

【0051】

ステップ402では、sQueuedMsgObjRef3021を参照して、sQueuedMsgObjA3022 のアドレスを決定する。

【0052】

次にステップ403の条件が成立する間、すなわち、上記決定したアドレスがNULLでない間、ステップ404からステップ411の手順を繰り返し、Data3008をFIFOBuffer(3029, 3037)へ書き込む。Data3008からFIFOBufferA3029 への書き込みを例に、ステップ404からステップ411を説明する。

【0053】

ステップ404では、FIFOBufferA3029があふれているか判定する。

【0054】

あふれている場合は、ステップ405で、Status3024に、FIFOBufferA3029があふれていることを示す値を書き込む。

【0055】

あふれていない場合は、まずステップ406で、WriteDataRef3026を参照して、Data3008をFIFOBufferA3029へ書き込む。

【0056】

次にステップ407で、WriteDataRef3026を更新する。

【0057】

ステップ408では、LIMIT4007 を参照して、バッファa2018があふれているか判定する。

【0058】

あふれている場合はステップ409で、Status3024に、バッファa2018があふれていることを示す値を書き込む。

【0059】

あふれていない場合はステップ410で、Status3024に、バッファa2018があふれていないことを示す値を書き込む。

【0060】

ステップ411では、sQueuedMsgObjRef3028を参照して、sQueuedMsgObjB3030のアドレスを決定する。

【0061】

図9は、図5のステップ105の処理詳細を示すフローチャートである。図9を用いて、ステップ105を説明する。

【0062】

まずステップ500で、TransferMode3015を参照して、Periodicalメッセージ送信を利用するか判定する。

【0063】

Periodicalメッセージ送信を利用する場合は、ステップ501でTimePeriod3016を参照して、Periodicalメッセージ送信を行う。

【0064】

Periodicalメッセージ送信を利用しない場合は、まずステップ502で、Size3002を参照して、送信するメッセージのデータサイズを決定する。

【0065】

次にステップ503で、DataRef3004を参照して、Data3008のアドレスを決定する。

【0066】

ステップ504では、Handle3017を参照して、送信するメッセージのメッセージIDを決定する。

【0067】

ステップ505では、CANドライバの送信処理2032を呼び出し、CAN

コントローラ 2003 を用いて、ネットワーク 2005 へメッセージを送信する。
この呼び出しにより、CAN ドライバの送信処理のタスクが起動されるが、その処理内容については後述する。

【0068】

ステップ 506 では、ステップ 505 のリターンパラメータから、Data6003 に送信するメッセージを書き込めたか判定する。

【0069】

書き込めた場合には、まずステップ 507 で、APa2012 が送信するメッセージのデータを、Data3008 に書き込む。

【0070】

次にステップ 508 で、Data3008 を FIFO Buffer (3029, 3037) へ書き込む。

【0071】

図 10 は、図 5 のステップ 106 の処理詳細を示すフローチャートである。図 10 を用いて、ステップ 106 を説明する。

【0072】

まずステップ 700 で、TransferMode3015 を参照して、Periodical メッセージ送信を利用するか判定する。

【0073】

Periodical メッセージ送信を利用する場合には、ステップ 701 で、TimePeriod3016 を参照して、Periodical メッセージ送信を行う。

【0074】

Periodical メッセージ送信を利用しない場合には、まずステップ 702 で、Size3002 を参照して、送信するメッセージのデータサイズを決定する。

【0075】

次にステップ 703 で、DataRef3004 を参照して、Data3008 のアドレスを決定する。

【0076】

ステップ 704 では、Handle3017 を参照して、送信するメッセージのメッセー

ジIDを決定する。

【0077】

ステップ705では、CANコントローラ2003を用いて、ネットワーク2005へメッセージを送信する。

【0078】

ステップ706では、ステップ705のリターンパラメータから、Data6003に送信するメッセージを書き込めたか判定する。

【0079】

書き込めた場合には、まずステップ707で、APa2012が送信するメッセージのデータを、Data3008に書き込む。

【0080】

次にステップ708で、Data3008をBuffer3009へ書き込む。

【0081】

次に、OSEK-COMのメッセージ受信処理2011について詳細に説明する。メッセージ受信処理2011は、各アプリケーションプログラムが、OSEK-COM2008を用いて実際のメッセージ受信を行うためのプログラムである。本実施例では、OSEK-COMのメッセージ受信処理は、それを呼び出したアプリケーションプログラムのタスクで実行されるものとする。図11は、メッセージ受信処理2011の流れを示すフローチャートである。図11を用いて、メッセージ受信処理2011について説明する。例えば、APa2012が、メッセージオブジェクトa2015、バッファa2018、メールボックスa2028を利用して、メッセージ受信を行うとする。また、APa2012が受信するメッセージは、CANコントローラ2003によって、メールボックスa2028へ格納され、CANドライバのメッセージ送信・受信完了割り込み処理2034によって、バッファa2018へ格納されているとする。

【0082】

まず、ステップ900で、APa2012は、メッセージ受信で利用するメッセージオブジェクトa2015を決定する。

【0083】

次にステップ901で、TransferDir3014を参照して、受信するメッセージが受信メッセージであるか判定する。

【0084】

受信メッセージの場合、まずステップ902で、DataRef3004を参照して、Data3008のアドレスを決定する。

【0085】

ステップ903では、Handle3017を参照して、受信するメッセージのメッセージIDを決定する。

【0086】

ステップ904は、CANドライバの受信処理2033を呼び出し、この呼び出しにより、CANドライバの受信処理のタスクが起動されるが、その処理内容については後述する。バッファa2018からData3008へ受信するメッセージのデータを書き込む。

【0087】

ステップ905では、ステップ904のリターンパラメータから、バッファa2018からData3008へ受信するメッセージのデータを書き込めたか判定する。

【0088】

書き込めた場合には、ステップ906で、受信するメッセージがUnqueuedメッセージの場合(3012)、Data3008をBuffer3009へ書き込む。

【0089】

受信するメッセージがQueuedメッセージの場合(3012)、Data3008をFIFOBuffer(3029, 3037)へ書き込む。

【0090】

最後にステップ907で、メッセージオブジェクトa2015からAPa2012へデータを読み込む。

【0091】

図12は、図11のステップ907の処理詳細を示すフローチャートである。

図 12 を用いて、ステップ 907 を説明する。

【0092】

まずステップ 1100 で、Size3002 を参照して、受信メッセージのデータサイズを決定する。

【0093】

次にステップ 1101 で、受信するメッセージが Queued メッセージであるか判定する。

【0094】

Queued メッセージである場合、まずステップ 1102 で、sQueuedMsgObjRef3021 を参照して、sQueuedMsgObjA3022 のアドレスを決定する。

次にステップ 1103 で、Status3024 を参照して、FIFOBufferA3029 に受信メッセージがあるか判定する。

【0095】

受信メッセージがある場合、まずステップ 1104 で、ReadDataRef3027 を参照して、FIFOBufferA3029 から APa2012 へ受信データを読み込む。

【0096】

次にステップ 1105 で、ReadDataRef3027 を更新する。

【0097】

そしてステップ 1106 で、FIFOBufferA3029 は空か判定する。

【0098】

空であればステップ 1107 で、Status3024 に、FIFOBufferA3029 は空であることを示す値を格納する。

【0099】

空でなければステップ 1108 で、Status3024 に、FIFOBufferA3029 は空でないことを示す値を格納する。

【0100】

ステップ 1101 の判定で、Queued メッセージでない場合、まずステップ 1109 で、DataRef3004、Size3002 を参照して、Buffer3009 の読み込みアドレスを決定する。

【0101】

次にステップ1110で、Buffer3009からAPa2012へ受信データを読み込む。

そしてステップ1111で、Status3007に、Buffer3009は空であることを示す値を格納する。

【0102】

(e) CANドライバ

次にCANドライバ2009について詳細に説明する。まずCANドライバの構成について説明する。CANドライバ2009は、図1に示すように、バッファ2031、メッセージ属性テーブル2021、メッセージ送信処理2032、メッセージ受信処理2033、メッセージ送信・受信完了割り込み処理2034で構成される。

【0103】

バッファ2031は、バッファa2018、バッファb2019、バッファc2020等の複数のバッファで構成される。バッファ2031を構成するバッファの数は、送信・受信メールボックス2023を構成するメールボックスの数だけ用意し、各バッファは、各メールボックスの専用バッファとする。例えば、メールボックスa2028の専用バッファをバッファa2018とした場合、バッファは、Data6003と同様の、メッセージのデータ格納領域である。

【0104】

図13は、メッセージ属性テーブル2021の詳細な構成図である。メッセージ属性テーブル2021は、メールボックス番号4001、MessageID4002、DataSize4003、QUEUED4004、LOCKED4005、NOMSG4006、LIMIT4007で構成される。メールボックス番号4001は、各メールボックスのメールボックス番号を記憶する領域である。MessageID4002は、各メールボックス番号ごとに、メールボックスに登録されたメッセージIDを記憶する領域である。DataSize4003は、各メールボックス番号ごとに、メールボックスに登録されたメッセージのデータサイズを記憶する領域である。MessageID4002、DataSize4003は、例えば、メールボックスa2028を構成するMessageID6001、DataSize6002と同様の記憶領域である。QUEUED4004は、各メールボックス番号ごとに、メッセージがQueuedメッセ

ージであるか、Unqueuedメッセージであるかを記憶する領域である。メッセージがQueuedメッセージである場合にビットセットされる。LOCKED4005は、各メールボックス番号ごとに、メールボックスに格納されたメッセージが、メッセージ送信要求を受けて、ネットワーク2005へのメッセージ送信を実行中であるか、メッセージ送信要求を受けておらず、ネットワーク2005へのメッセージ送信を行っていないかを記憶する領域である。LOCKED4005は、メールボックスに格納されたメッセージが、メッセージ送信要求を受けて、ネットワーク2005へのメッセージ送信を実行中である場合にビットセットされる。NOMSG4006 は、各メールボックス番号ごとに、バッファ2031に受信メッセージが貯まっているか、受信メッセージが貯まっていないかを記憶する領域である。NOMSG4006 は、バッファ2031に受信メッセージが貯まっていない場合にビットセットされる。LIMIT4007 は、各メールボックス番号ごとに、バッファ2031に受信メッセージが限度まで貯まっているか、受信メッセージが限度まで貯まっていないかを記憶する領域である。LIMIT4007 は、バッファ2031の各バッファに受信メッセージが限度まで貯まっている場合にビットセットされる。本実施例ではQueuedメッセージ、Unqueuedメッセージを扱うため、例えば、バッファ2031にQueuedメッセージをキューイングするFIFOバッファ等を設ける必要がある。

【0105】

Unqueuedメッセージは、キューイングが不要なメッセージである。バッファ2031の各バッファをFIFOバッファとして実装した場合、FIFOバッファあふれの際にLIMIT4007 はビットセットされる。

【0106】

次にCANDライバの動作について説明する。CANDライバの処理は、アプリケーションプログラムやOSEK-COMとは別のタスクで実行される。まずCANDライバのメッセージ送信処理2032について詳細に説明する。本実施例では、OSEK-COMが、図9のステップ505により、CANDライバのメッセージ送信処理を呼び出すと、CANDライバのメッセージ送信処理タスクが起動される。CANDライバのメッセージ送信処理の呼び出しが複数発生すると、それらに対応して複数のタスクが起動される。

【0107】

図14は、CANドライバのメッセージ送信処理2032の処理詳細を示すフローチャートである。図14を用いて、CANドライバの送信処理を説明する。

まずステップ600で、メッセージ属性テーブル2021を用いて、送信するメッセージのメッセージIDから、メールボックス番号を決定する。

【0108】

次にステップ601で、LOCKED4005にビットセットされているか判定する。ビットセットされている場合は、ステップ602で、リターンパラメータに、Data6003に送信するメッセージを書き込めなかったことを示す値を格納する。

【0109】

ビットセットされていない場合は、まずステップ603で、LOCKED4005にビットセットする。

【0110】

次にステップ604で、APa2012が送信するメッセージのデータをData6003に書き込む。

【0111】

そしてステップ605で、TXPR5002にビットセットして、CANコントローラ2003にネットワーク2005へのメッセージ送信を要求する。

【0112】

ステップ606では、リターンパラメータに、Data6003に送信するメッセージを書き込めたことを示す値を格納する。

【0113】

以上の処理が終了すると、CANドライバのメッセージ送信タスクは終了する。

【0114】

次にCANドライバのメッセージ受信処理2033について詳細に説明する。本実施例では、OSEK-COMが図11のステップ904によりCANドライバのメッセージ受信処理を呼び出すと、CANドライバの受信処理タスクが起動される。CANドライバのメッセージ受信処理の呼び出しが複数発生すると、そ

れらに対応して複数のタスクが起動される。

【0115】

図15は、CANドライバのメッセージ受信処理2033の処理詳細を示すフローチャートである。図15を用いて、CANドライバの受信処理を説明する。

まずステップ1000で、メッセージ属性テーブル2021を用いて、受信するメッセージのメッセージIDから、メールボックス番号を決定する。

【0116】

ステップ1001では、MBIMR5005にビットセットする。

【0117】

ステップ1002では、NOMSG4006にビットセットされているか判定する。

【0118】

ステップ1002の判定においてビットセットされている場合は、ステップ1003で、リターンパラメータに、バッファa2018に受信するメッセージが書き込まれていないことを示す値を格納する。

【0119】

ステップ1002の判定においてビットセットされていない場合は、まずステップ1004で、LIMIT4007にビットセットされているか判定する。

【0120】

ステップ1004の判定においてビットセットされている場合は、まずステップ1005で、リターンパラメータには、バッファa2018があふれていることを示す値を格納する。

【0121】

次にステップ1006で、LIMIT4007をクリアする。

【0122】

ステップ1004の判定においてビットセットされていない場合は、ステップ1007で、リターンパラメータには、バッファa2018があふれていないことを示す値を格納する。

【0123】

そしてステップ1008では、バッファa2018からData3008へ受信するメ

メッセージのデータを書き込む。

【0124】

ステップ1009では、NOMSG4006 にビットセットする。

【0125】

ステップ1010では、MBIMR5005 をクリアする。

【0126】

以上の処理が終了すると、CANドライバのメッセージ受信タスクは終了する。

【0127】

次に、CANドライバのメッセージ送信・受信完了割り込み処理2034について詳細に説明する。図16は、メッセージ送信・受信完了割り込み処理2034の処理詳細を示すフローチャートである。図16を用いて、メッセージ送信・受信完了割り込み処理について説明する。

【0128】

まずステップ800で、CANコントローラ2003は、メッセージ送信完了割り込みか判定する。

【0129】

メッセージ送信完了割り込みである場合は、ステップ801の条件が成立する間、すなわち、CANコントローラ2003のメールボックス数だけ、メールボックス番号の小さいものから順番に、ステップ802からステップ804の手順を繰り返して実行する。

【0130】

ステップ802で、TXACK5003 にビットセットされているか判定する。

【0131】

ビットセットされている場合は、まずステップ803で、LOCKED4005をクリアする。

【0132】

次にステップ804で、TXACK5003 をクリアする。

【0133】

ステップ805では、メッセージ送信完了割り込みフラグをクリアする。

【0134】

ステップ806で、CANコントローラ2003は、メッセージ受信完了割り込みか判定する。

【0135】

メッセージ受信完了割り込みである場合は、ステップ807の条件が成立する間、すなわち、CANコントローラ2003のメールボックス数だけ、メールボックス番号の小さいものから順番に、ステップ808からステップ815の手順を繰り返して実行する。

【0136】

ステップ808で、RXPR5004にビットセットされているか判定する。

【0137】

ビットセットされている場合は、ステップ809で、QUEUED4004にビットセットされているか判定する。

【0138】

ステップ809の判定で、ビットセットされている場合は、ステップ810で、NOMSG4006にビットセットされているか判定する。

【0139】

ステップ810の判定で、ビットセットされている場合は、ステップ811で、メールボックスa2028からバッファa2018へ受信メッセージを書き込む。

【0140】

ステップ810の判定で、ビットセットされていない場合は、ステップ812で、LIMIT4007にビットセットする。

【0141】

ステップ809の判定で、ビットセットされていない場合は、ステップ813で、メールボックスa2028からバッファa2018へ受信メッセージを書き込む。

【0142】

ステップ814では、NOMSG4006をクリアする。

【0143】

ステップ815では、RXPR5004をクリアする。

【0144】

(f) OS

次にOS2050について詳細に説明する。OS2050は、タスク優先順位管理テーブル2051とタスクのスケジューリングを行うスケジューラ2052を持つ。タスク優先順位管理テーブル2051を用いたスケジューラ2052のタスクスケジューリングの方法について説明する。

【0145】

図17は、タスク優先順位管理テーブルの構成図である。図17を用いて、タスク優先順位管理テーブルの構成について説明する。タスク優先順位管理テーブルは、メッセージのメッセージID1200とタスク優先順位1201で構成される。例えば、メッセージIDの優先順位が100, 200, 300の順で高いとすると、メッセージIDの優先順位に対応して、そのメッセージを処理するタスクの優先順位も1, 2, 3の順で高くなるように、タスク優先管理テーブルをあらかじめ設定しておく。スケジューラ2052は、タスク優先順位管理テーブル2051を参照して、タスク優先順位が高いタスクを優先的に起動する。

【0146】

本実施例では、OSEK-COM2008からの呼び出しにより、CANドライバ2009のメッセージ送信処理2032やメッセージ受信処理2033のタスクが起動される。CANドライバ2009のメッセージ送信処理2032のタスク起動により、送信・受信メールボックス2023に送信メッセージを格納して、CANコントローラ2003にネットワーク2005への送信要求ができるとする。

【0147】

メッセージIDが300のメッセージを送信する、CANドライバのメッセージ送信処理タスクを起動する場合、スケジューラ2052は、タスク優先順位管理テーブル2051を参照して、そのタスクを優先順位3で起動する。起動され

たタスクは、CANドライバ2009を用いて、送信・受信メールボックス2023に送信メッセージを格納して、CANコントローラ2003にネットワーク2005への送信を要求する。

【0148】

メッセージIDが300のメッセージを送信する、CANドライバのメッセージ送信処理タスクが起動中に、メッセージIDが100のメッセージを送信する、CANドライバのメッセージ送信処理タスクを後から起動する場合、スケジューラ2052は、タスク優先順位管理テーブル2051を参照して、そのタスクを優先順位1で起動すると同時に、優先順位3で起動中のタスクを一時待機させる。これにより、メッセージIDが300のメッセージより先に、メッセージIDが100のメッセージを送信・受信メールボックス2023に格納して、CANコントローラ2003にネットワーク2005への送信を要求できる。

【0149】

CANドライバのメッセージ受信処理タスクも同様に、受信したメッセージIDに応じた優先順位で起動される。

【0150】

本発明によれば、あるCANドライバのメッセージ送信処理タスクが、CANコントローラのメールボックスへメッセージを格納して、ネットワークへのメッセージ送信要求を実行中であっても、後から起動された別のCANドライバのメッセージ送信処理タスクが、優先順位の高いメッセージをCANコントローラのメールボックスへ格納して、ネットワークへのメッセージ送信要求を実行することができる。これにより、最高優先順位のメッセージを常に最高優先順位で送信することができ、メッセージ送信のリアルタイム性を向上させることができる。

また本発明によれば、あるCANドライバのメッセージ受信処理タスクが、CANコントローラのメールボックスからOSEK-COMのメッセージオブジェクトへメッセージを格納中であっても、後から起動された別のCANドライバのメッセージ受信処理タスクが、優先順位の高いメッセージをCANコントローラのメールボックスからOSEK-COMのメッセージオブジェクトへメッセージを格納することができる。これにより、最高優先順位のメッセージを常に最高

優先順位で受信することができ、メッセージ受信のリアルタイム性を向上させることができる。

【0151】

(2) 本発明の他の実施例

これまでの説明では、CANドライバに、受信メッセージを一時記憶するバッファを設けたが、バッファを設けずにCANコントローラから受信メッセージを直接OSEK-COMのメッセージオブジェクトに書き込んでもよい。これにより、メッセージ受信処理実行時間の削減やメモリ資源の節約ができる。

【0152】

また、複数のメールボックスを有するCANコントローラを用い、各メールボックスには特定のメッセージIDを1つだけ割り当てたが、1つのメールボックスに複数のメッセージIDを割り当てることもできる。これにより、CANコントローラが有するメールボックスの数に関係なく、メッセージIDの異なる種類のメッセージを送信、受信することができる。

【0153】

更に、CANドライバに、受信メッセージを一時記憶するバッファを1つのメールボックスに対して1つずつ設けたが、1つのメールボックスに対して複数のバッファを設けることができる。これにより、1つのメールボックスに複数のメッセージを受信した場合、受信メッセージのメッセージIDごとにバッファリングすることができ、メッセージの優先順位に応じてCANドライバのメッセージ受信処理タスクを起動することで、メッセージ受信のリアルタイム性を向上させることができる。

【0154】

また、CANドライバに、受信メッセージを一時記憶するバッファを設けたが、送信メッセージを一時記憶するバッファを設けてもよい。またバッファは、1つのメールボックスに対して複数のバッファを設けてもよい。これにより、1つのメールボックスを用いて複数のメッセージを送信する場合、送信メッセージのメッセージIDごとにバッファリングすることができ、メッセージの優先順位に応じてCANドライバのメッセージ送信処理タスクを起動することで、メッセー

ジ送信のリアルタイム性を向上させることができる。

【0155】

CANプロトコルを使用したか、他の通信プロトコルを使用してもよい。これにより、CANプロトコルだけでなく、他のプロトコルへ対応できる。

【0156】

通信ライブラリとしてOSEK-COMを使用したか、メッセージ送信処理、メッセージ受信処理を行う他の通信ライブラリを使用してもよい。これにより、OSEK-COMだけでなく、他の通信ライブラリへ対応できる。

【0157】

【発明の効果】

本発明によれば、最高優先順位のメッセージを常に最高優先順位で送信することができ、メッセージ送信のリアルタイム性を向上させる効果がある。

【図面の簡単な説明】

【図1】

リアルタイム分散システムの構成図である。

【図2】

リアルタイム分散システムの制御ユニット中のCANコントローラを構成する、メールボックスの詳細な構成図である。

【図3】

CANコントローラの、コントロールレジスタの詳細な構成図である。

【図4】

リアルタイム分散システムの制御ユニット中のOSEK-COMを構成する、メッセージオブジェクトの詳細な構成図である。

【図5】

メッセージ送信処理の流れを示すフローチャートである。

【図6】

メッセージ送信処理の、メッセージオブジェクトのデータおよびバッファ更新の流れを示すフローチャートである。

【図 7】

メッセージ送信処理の、メッセージオブジェクトのバッファ更新の流れを示すフローチャートである。

【図 8】

メッセージ送信処理の、メッセージオブジェクトの F I F O バッファ更新の流れを示すフローチャートである。

【図 9】

メッセージ送信処理の、Queued メッセージの送信の流れを示すフローチャートである。

【図 10】

メッセージ送信処理の、Unqueued メッセージの送信の流れを示すフローチャートである。

【図 11】

メッセージ受信処理の流れを示すフローチャートである。

【図 12】

メッセージ受信処理の、メッセージオブジェクトからのメッセージ受信の流れを示すフローチャートである。

【図 13】

リアルタイム分散システムの制御ユニット中の C A N ドライバを構成する、メッセージ属性テーブルの詳細な構成図である。

【図 14】

ローラによるメッセージ送信の流れを示すフローチャートである。

【図 15】

C A N ドライバ内バッファからのメッセージ受信の流れを示すフローチャートである。

【図 16】

メッセージ送信・受信完了割り込み処理の流れを示すフローチャートである。

【図 17】

リアルタイム分散システムの制御ユニット中の O S を構成する、タスク優先順

位管理テーブルの詳細な構成図である。

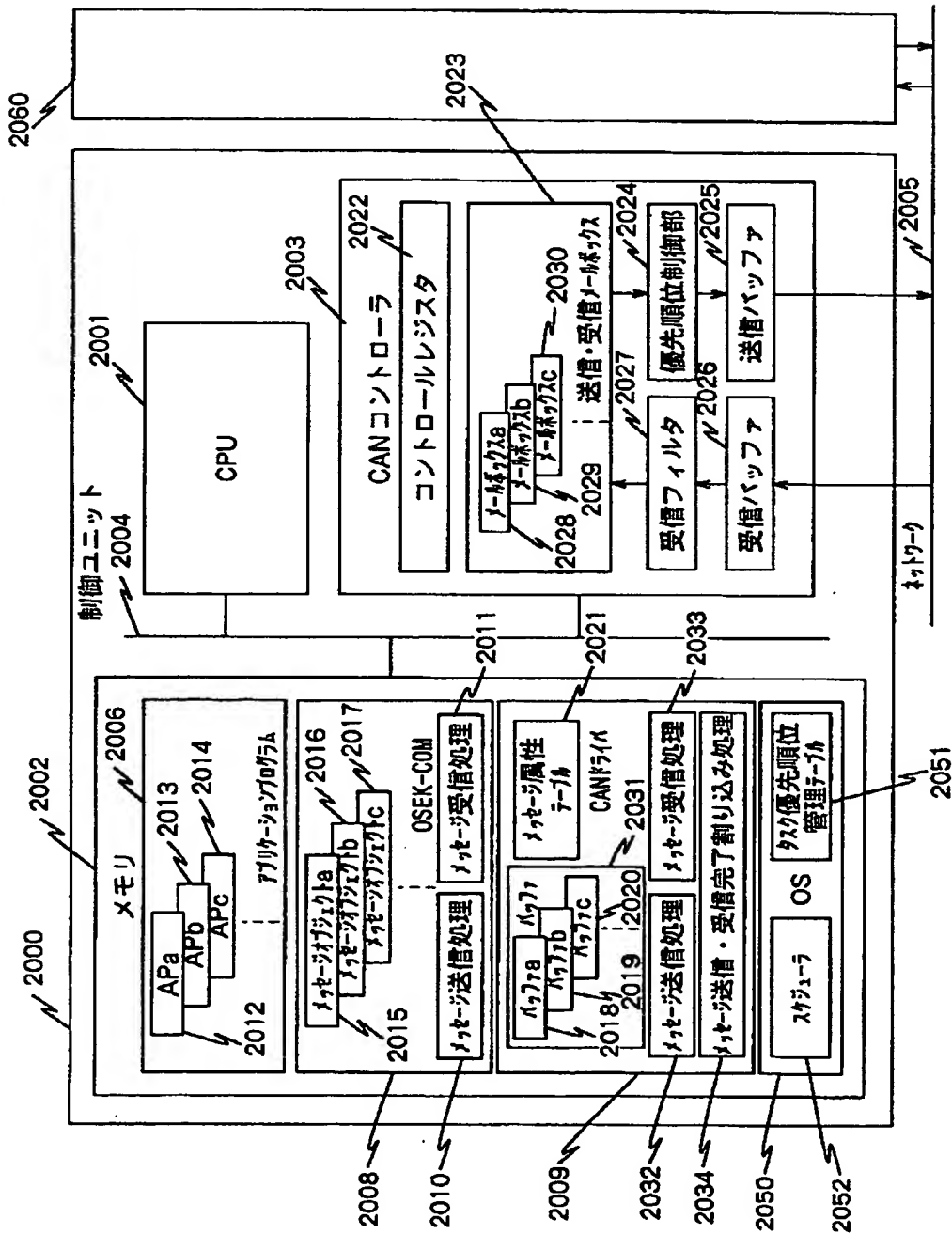
【符号の説明】

2000…制御ユニット、2001…CPU、2002…メモリ、2003…CANコントローラ、2004…バス、2005…ネットワーク、2006…アプリケーションプログラム、2008…OSEK-COM、2009…CANDライバ、2010…OSEK-COMのメッセージ送信処理、2011…OSEK-COMのメッセージ受信処理、2021…メッセージ属性テーブル、2022…コントロールレジスタ、2023…送信・受信メールボックス、2024…優先順位制御部、2025…送信バッファ、2026…受信バッファ、2027…受信フィルタ、2031…バッファ、2032…CANDライバのメッセージ送信処理、2033…CANDライバのメッセージ受信処理、2034…CANDライバのメッセージ送信・受信完了割り込み処理、2050…OS、2051…タスク優先順位管理テーブル、2052…スケジューラ。

【書類名】 図面

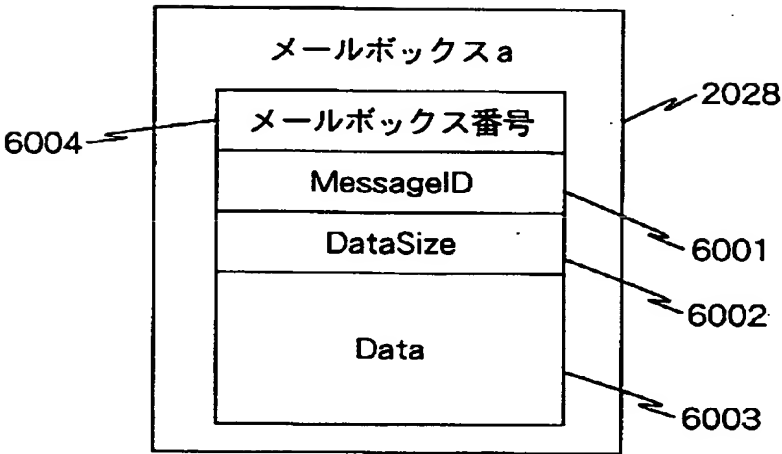
【図 1】

図 1



【図 2】

図 2



【図 3】

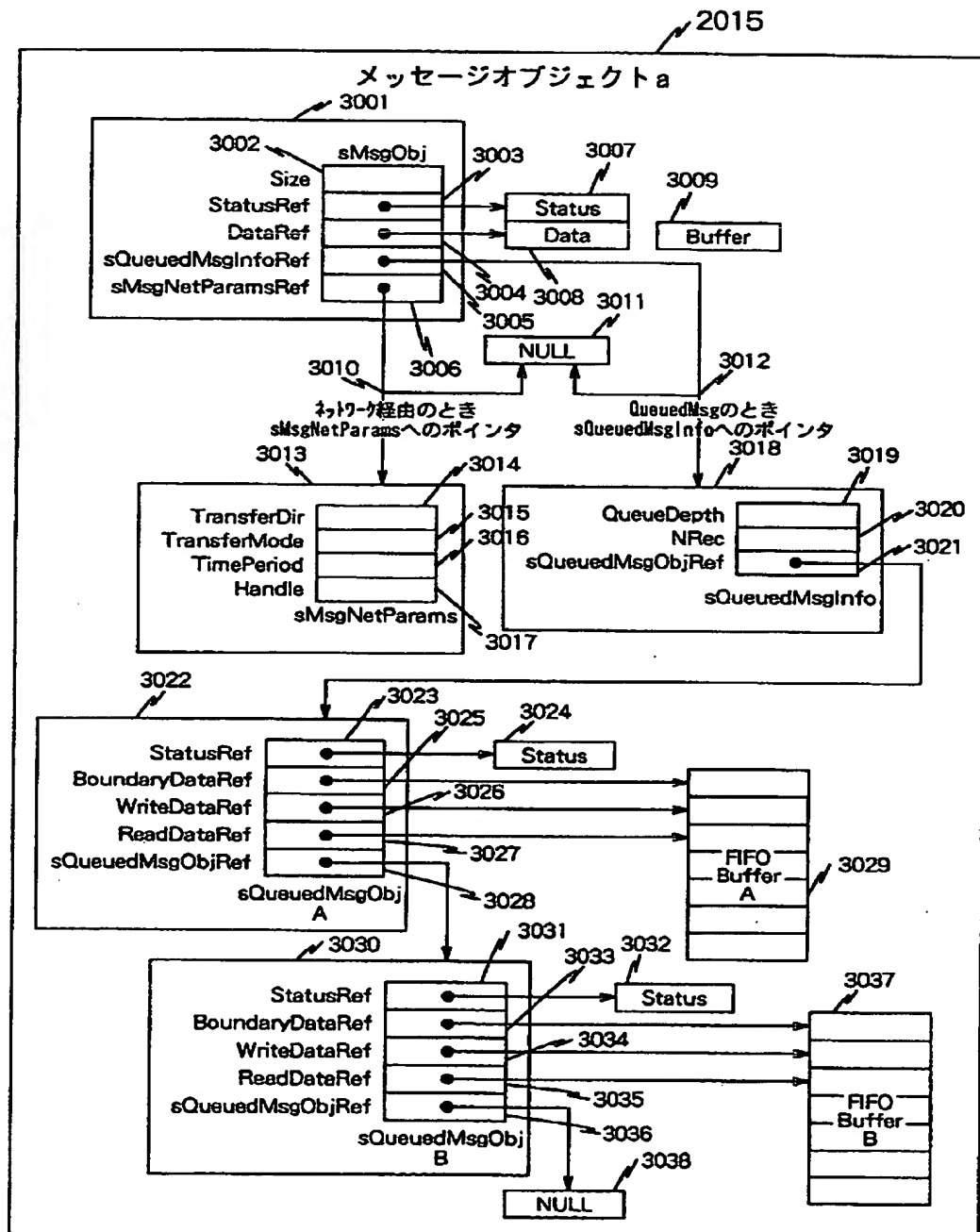
図 3

2022

コントロールレジスタ				
メールボックス番号	TXPR	TXACK	RXPR	MBIMR
0	1	0	0	0
1	0	0	1	0
2	0	0	0	0
・	・	・	・	・
・	・	・	・	・

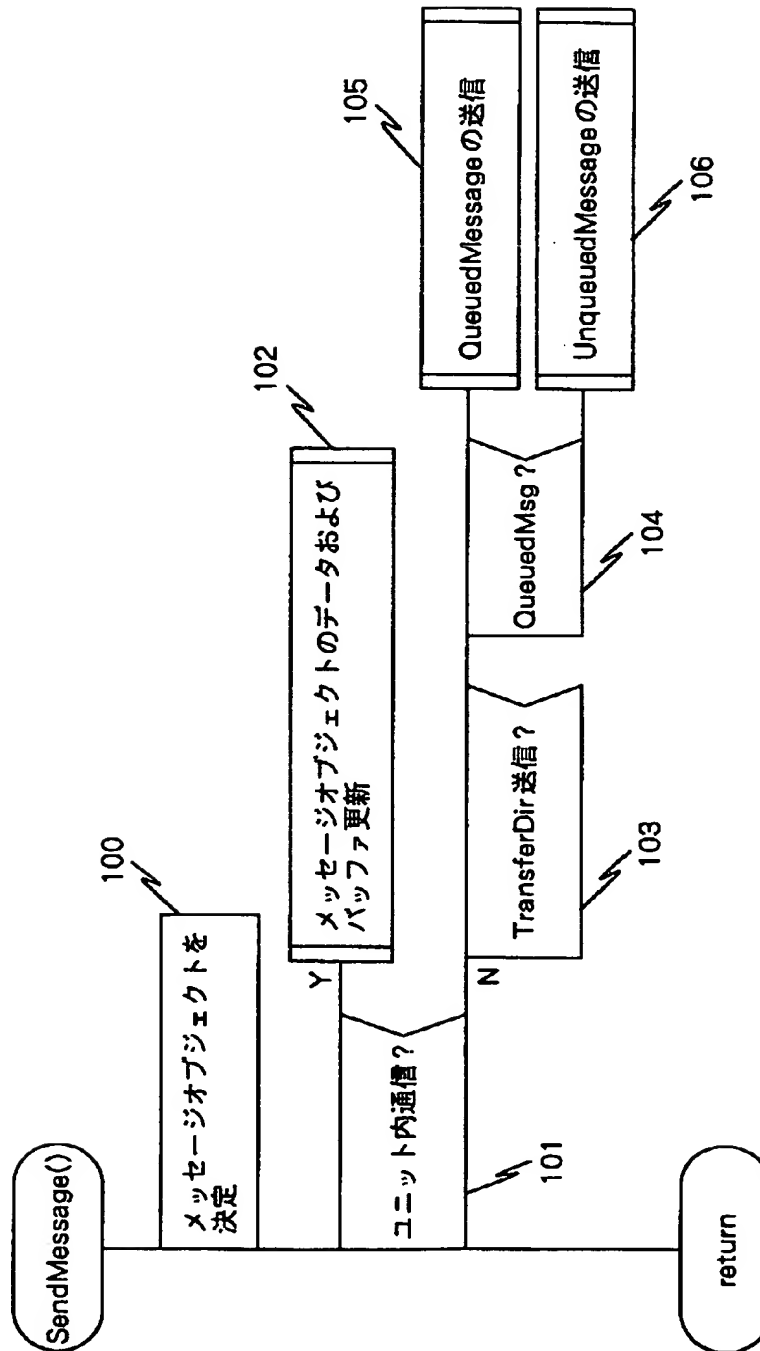
【図 4】

図 4



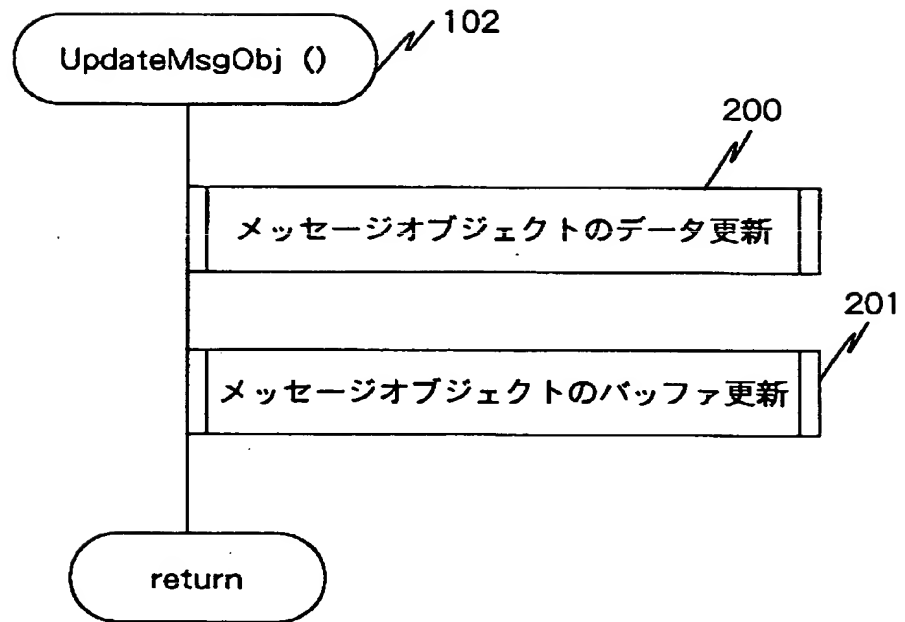
【図 5】

図 5



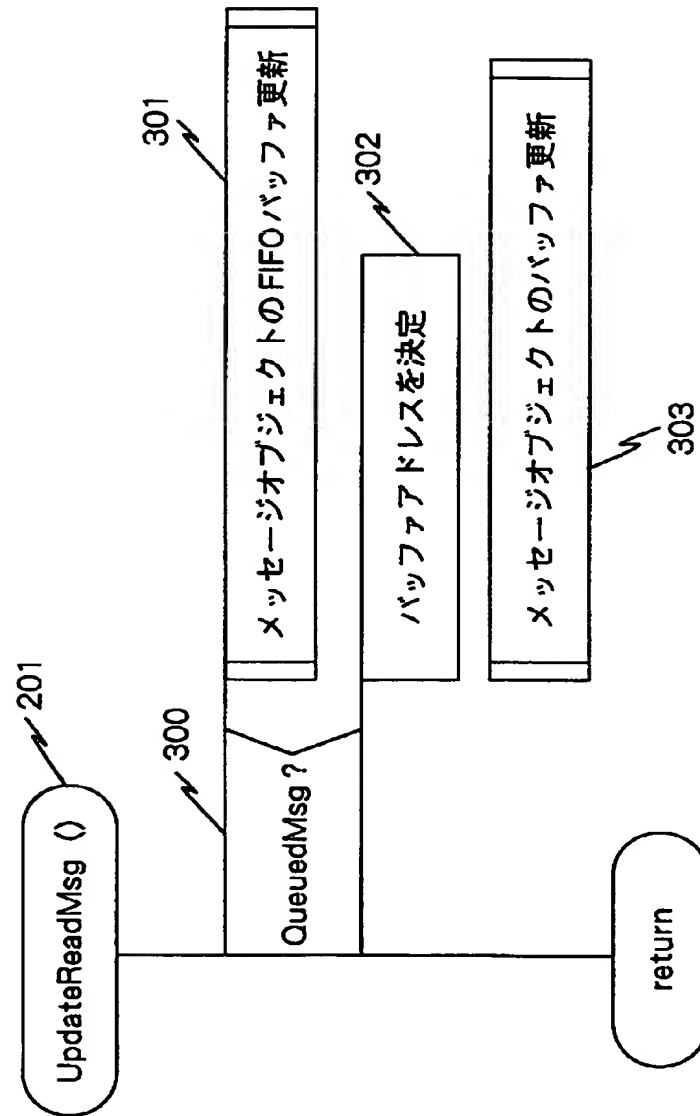
【図 6】

図 6

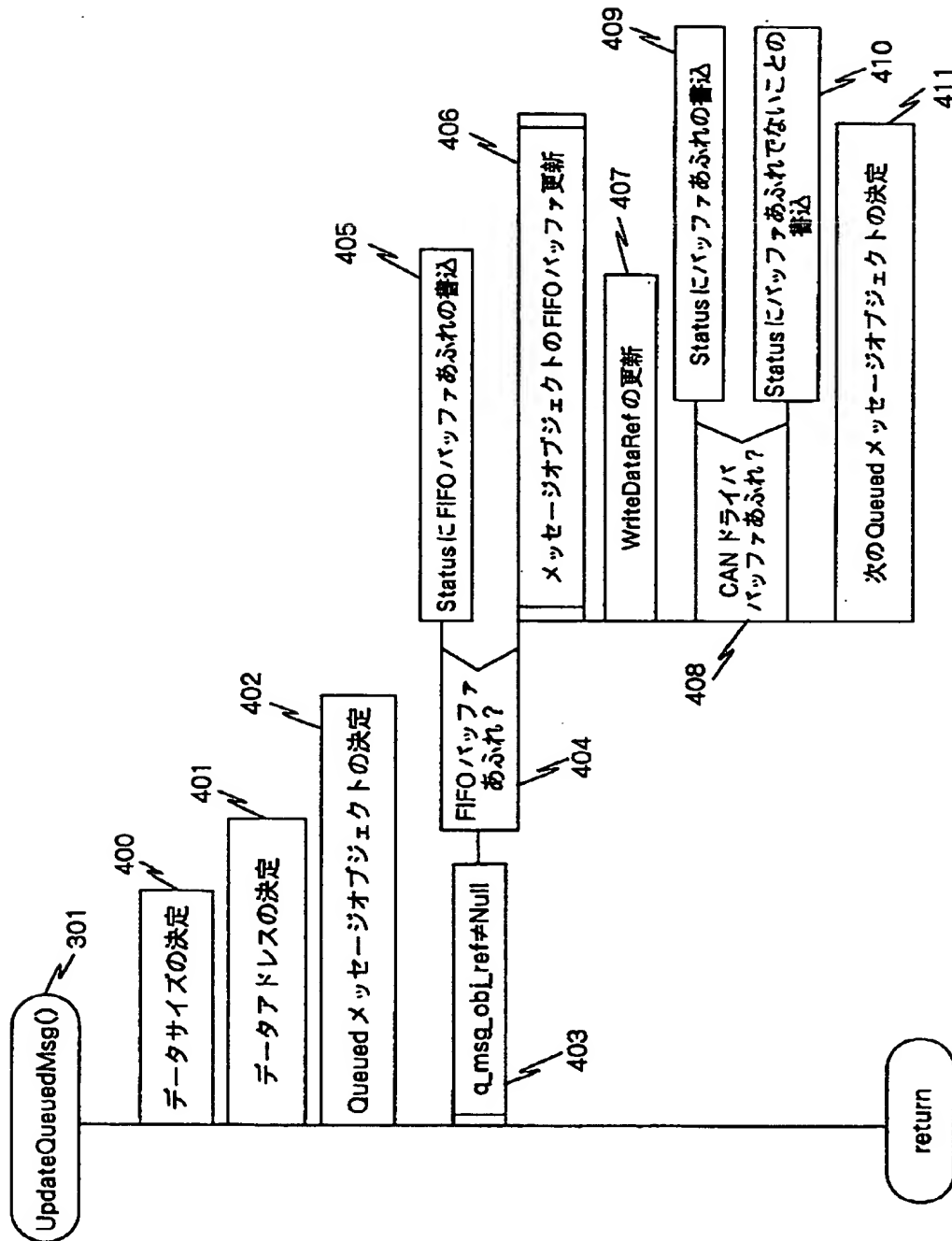


【図 7】

図 7

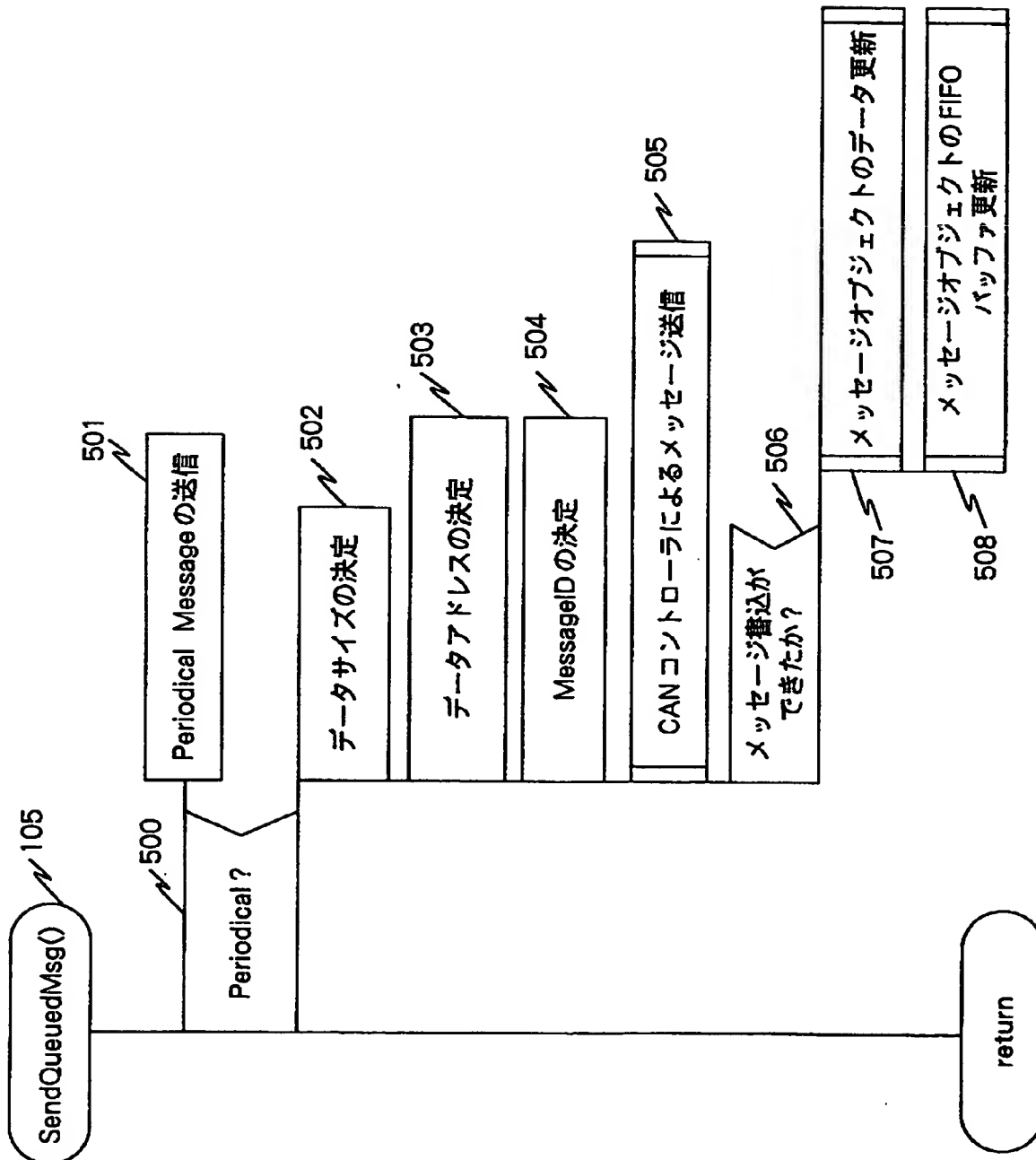


【図 8】



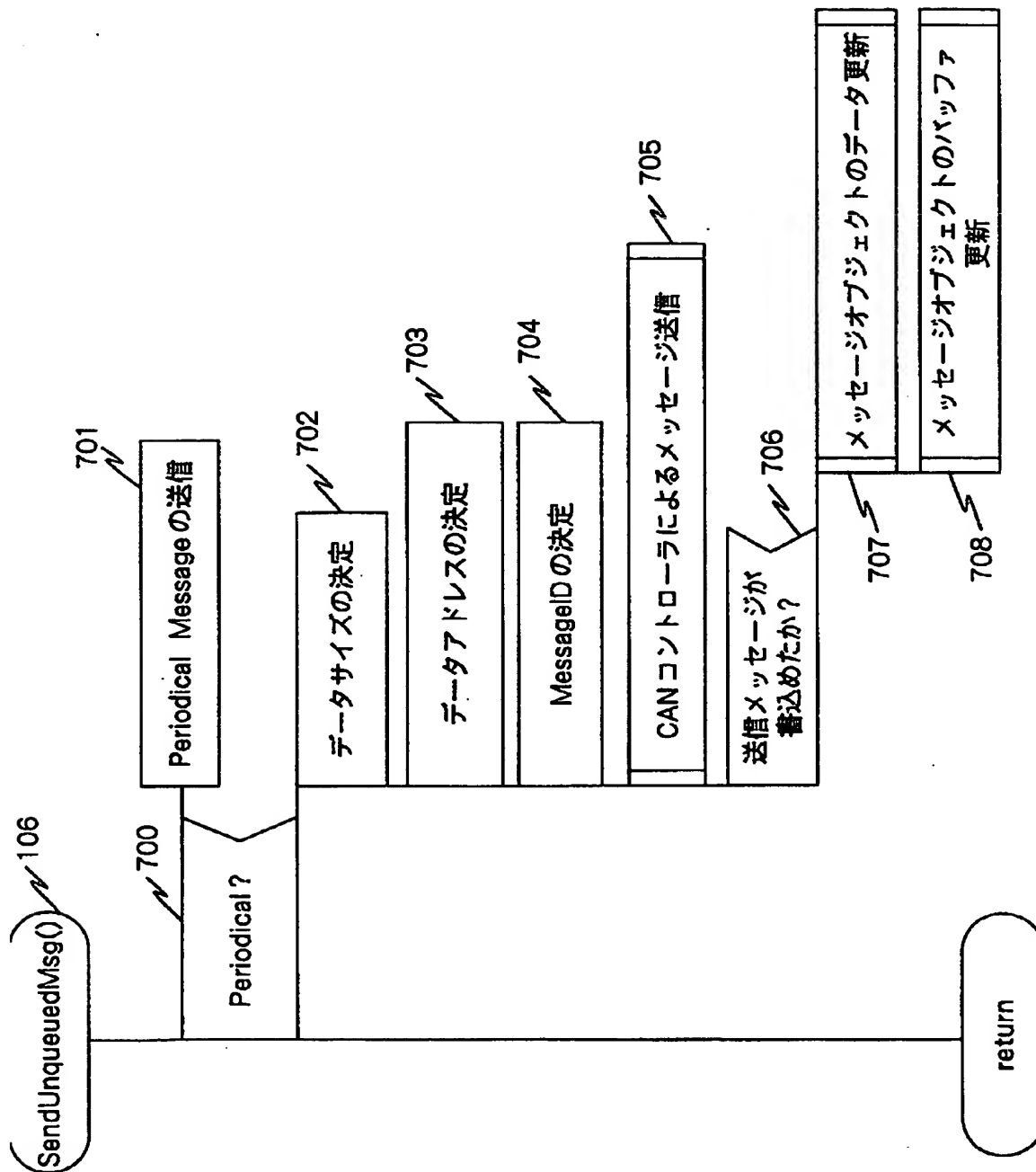
【図 9】

図 9



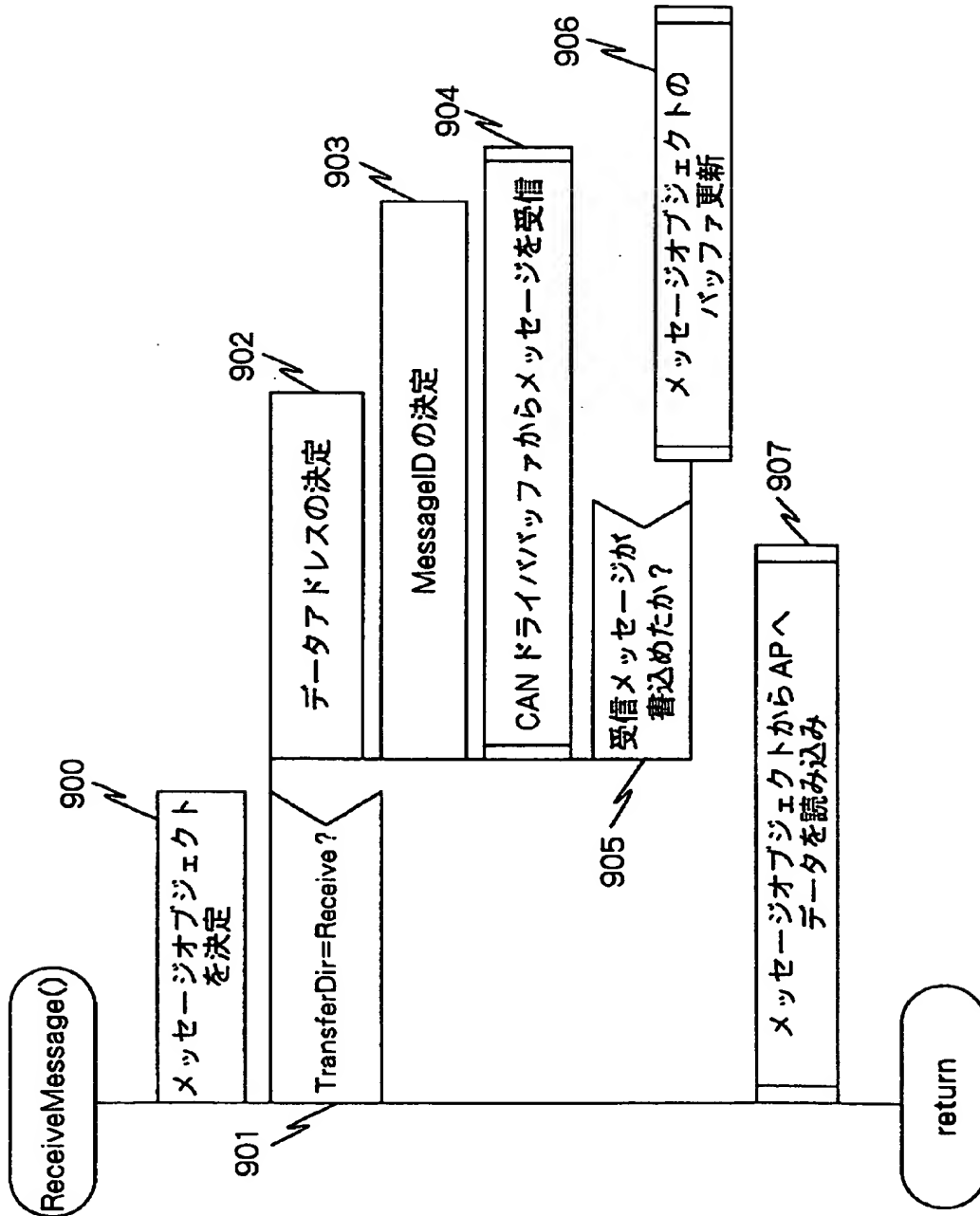
【図 10】

図 10



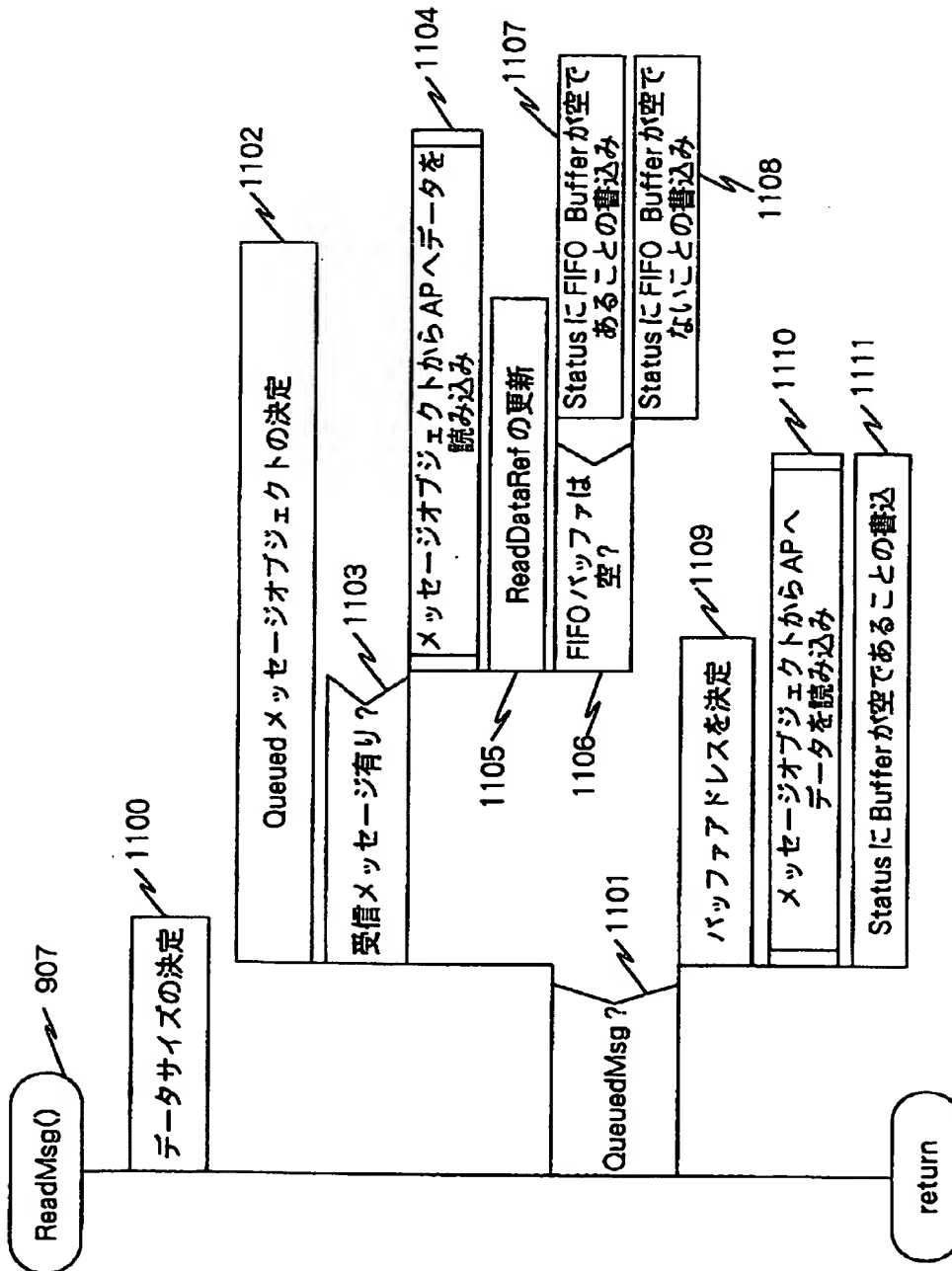
【図 11】

図 11



【図 12】

図 12



【図 13】

図 13

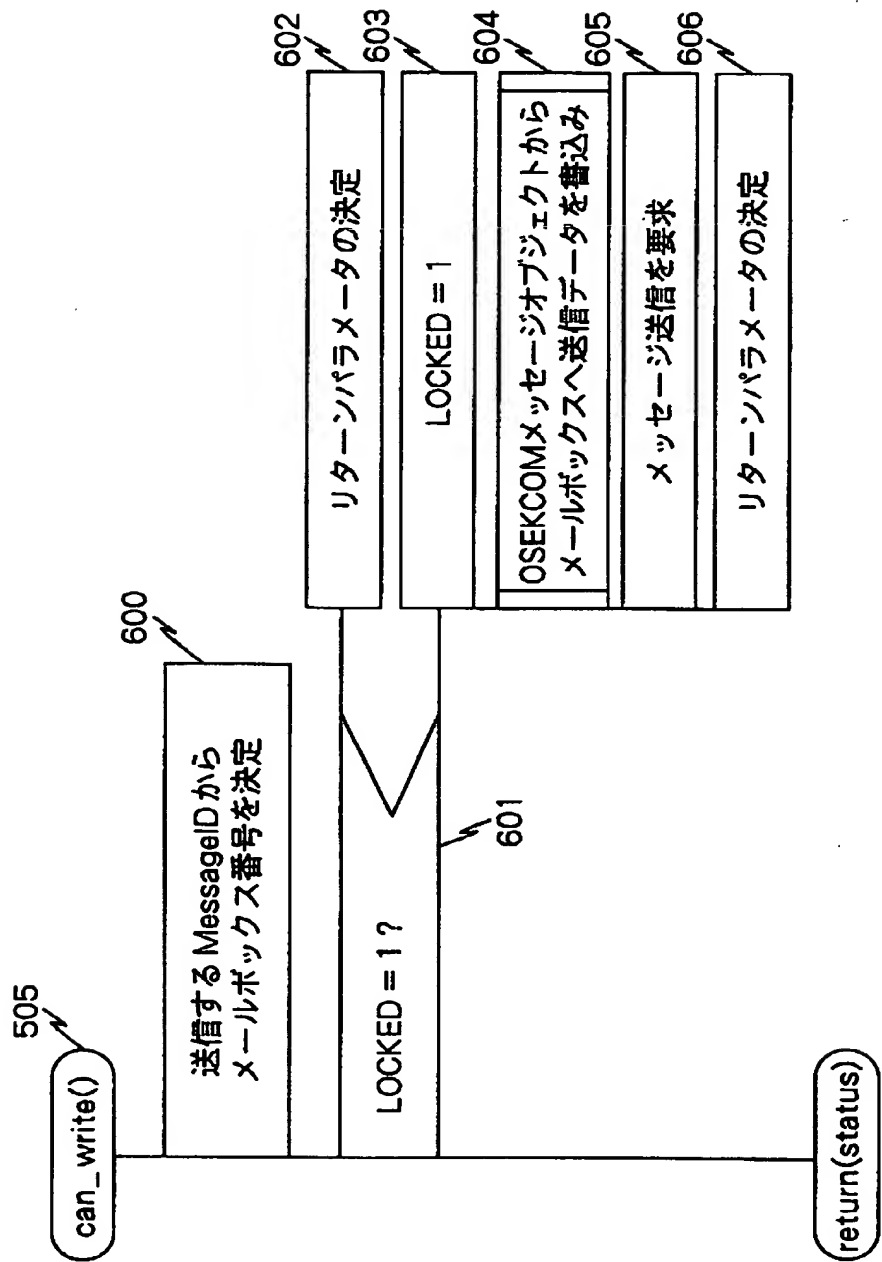
2021
N

メッセージ属性テーブル

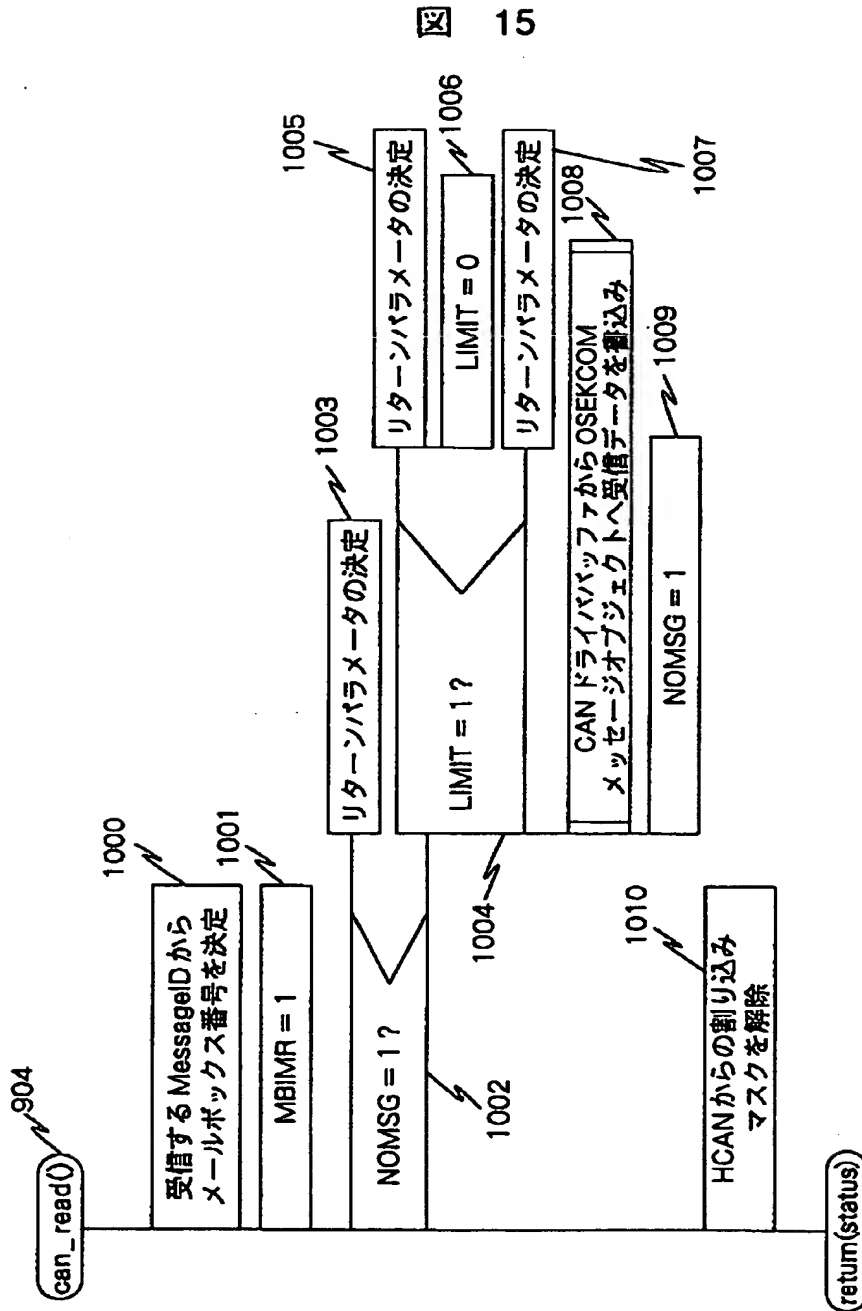
メッセージ番号	MessageID	DataSize	QUEUED	LOCKED	NOMSG	LIMIT
0	100	8	0	1	1	0
1	200	8	1	0	1	0
2	500	8	0	0	1	0
.
.

【図 1 4】

図 14

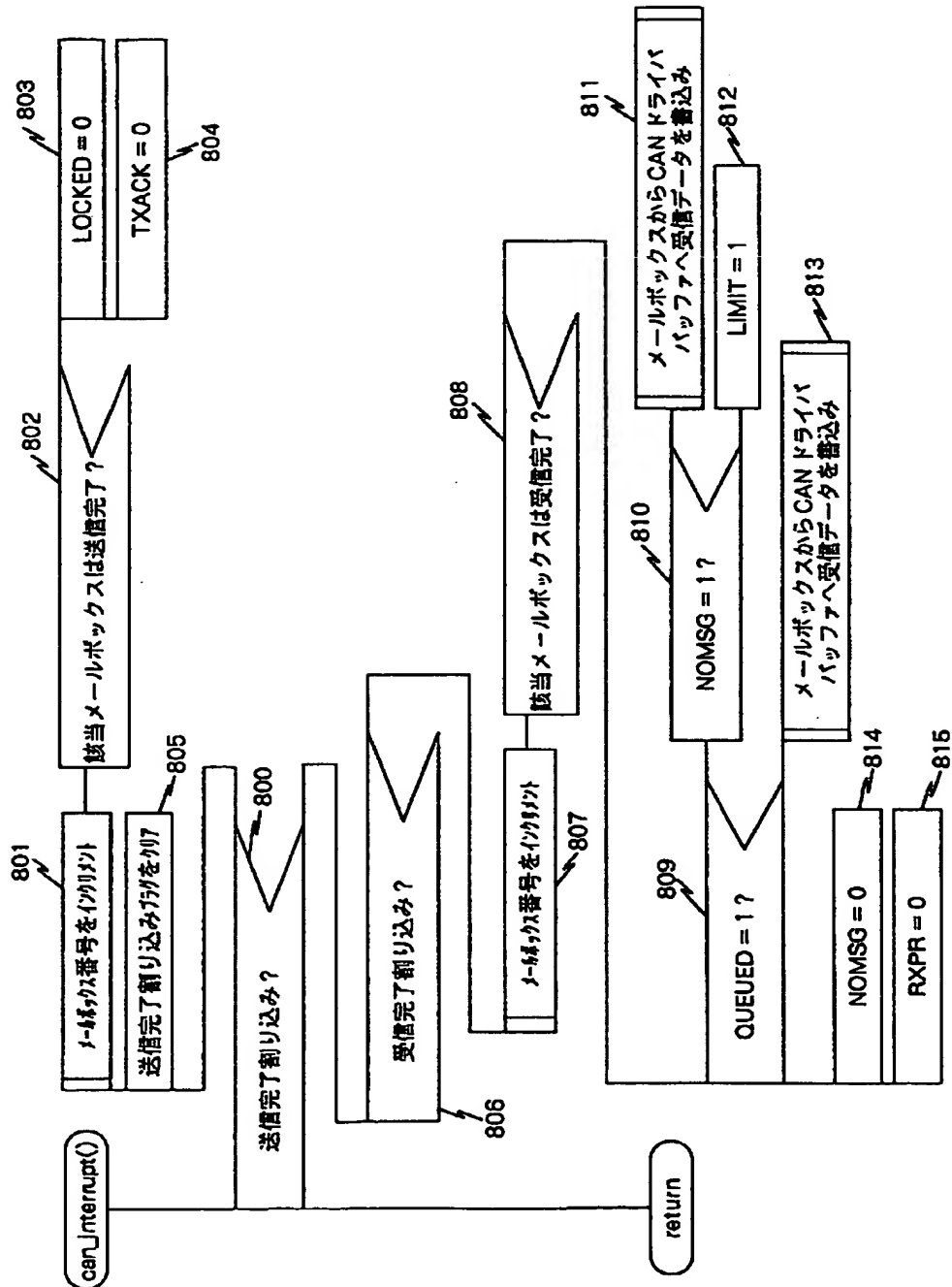


【図 15】



【図 16】

図 16



【図 17】

図 17

2051

タスク優先順位管理テーブル	
1200	1201
メッセージ ID	タスク優先順位
100	1
200	2
300	3

【書類名】 要約書

【要約】

【課題】

送受信要求のなされたメッセージをその優先順位に応じて送受信できるリアルタイム分散システムを提供する。

【解決手段】

本発明では、複数の送受信メッセージを格納してネットワーク通信を行うネットワークコントローラと、該ネットワークコントローラを用いてネットワーク通信を行うネットワークドライバと、扱う送受信メッセージの優先順位に応じてネットワークドライバの処理の優先順位を決定するネットワークドライバ優先順位管理手段と、ネットワークドライバの処理の優先順位にしたがってネットワークドライバの処理を実行するスケジューリング手段を設ける。

【選択図】 図1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【提出日】 平成10年 9月21日

【特許出願人】

【識別番号】 000005108

【住所又は居所】 東京都千代田区神田駿河台四丁目 6 番地

【氏名又は名称】 株式会社日立製作所

【代理人】 申請人

【識別番号】 100068504

【住所又は居所】 東京都千代田区丸の内 1 - 5 - 1 株式会社日立製
作所 知的所有権本部内

【氏名又は名称】 小川 勝男

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所